

The World Islamic Sciences and Education University

جامعة العلوم الإسلامية العالمية

Faculty of Information Technology

كلية تكنولوجيا المعلومات



GRADUATION PROJECT

Title

Course Withdrawal System

Students

Layan Al-Hassoon 3200605025

Aseel Al-Naami 3210605014

Supervisor

Dr. Abdullah Al-Zaqebah

SEMESTER II

2023/2024

Acknowledgments

We extend our heartfelt appreciation to Dr. Abdullah Zaqebah, our supervisor, for his unwavering guidance, support, and motivation during our graduation project. His exceptional expertise and invaluable perspectives significantly contributed to our project's success, and we deeply appreciate his consistent availability and commitment, even amidst his busy schedule. Our endeavor was marked by challenges that encouraged us to engage in critical thinking and push ourselves to attain our utmost potential. Dr. Abdullah Zaqebah's mentorship has been invaluable, and we express genuine gratitude for his guidance. Furthermore, we are thankful to the Software Engineering Department for furnishing us with the necessary resources and backing that enabled us to bring our project to completion.

Abstract

The development of an E-system for course withdrawal procedure at WISE University becomes an essential part of the WISE student information system, which improves the efficiency, and accessibility of the university's services. It aims to simplify the process of withdrawing from courses by transitioning from the traditional system to an automated system. The university can guarantee that withdrawal requests are processed accurately, on time, and error-free. This makes it easier to manage data and retain records, which helps to create a learning environment that is more responsive and organized. The traditional paper-based procedure consumes a lot of valuable time and effort for the students, instructors, and administrative staff. In addition to that, once the withdrawal request is submitted, there is no available method to track it. The Course Withdrawal System has made a significant change by eliminating the challenges that were addressed traditionally through automating the course withdrawal process, also providing the students with the ability to track their requests and receive notifications with every update on their requests. Key features include a secure login for authentication, a detailed withdrawal request form, real-time status tracking, and automated notifications. The proposed Course Withdrawal System will be integrated with the university's existing student information system (SIS), ensuring data consistency and accuracy. Therefor, for the compatibility issues, the PHP programming language is used to implement the server-side code (back-end) with the aid of MySQL database for storing the data. And for client-side (front-end) we adopt HTML, JavaScript, CSS, and Ajax for development. An interview and questionnaire are held to evaluate our system, results show high acceptance and encouragement of the idea.

In summary, WISE University's course withdrawal system aims to improve student satisfaction, minimize paperwork, and preserve academic integrity by offering an unambiguous, efficient, and easily accessible withdrawal process.

Contents

Acknowledgments.....	II
Abstract	III
List of Tables	VI
List of Figures	VII
List of abbreviations	VIII
CHAPTER 1	1
INTRODUCTION	1
1.1 Overview	2
1.2 Problem Statement	2
1.3 Project objectives	3
1.4 Research Strategy	3
1.5 Scope (boundary)	5
1.6 Gantt chart	5
1.6 Project outline	6
CHAPTER 2	7
2.1 Overview	8
2.2 Related Work	8
2.2.1 Withdrawal system automation	8
2.2.2 Challenges in the trsditional system	8
2.2.3 Current service vs. Automated service	9
2.3 Summary	11
CHAPTER 3	12
METHODOLOGY	12
3.1. Overview	13
3.2. Feasibility Study.....	13
3.2.1 Technical feasibility.....	13
3.2.2 Operational feasibility.....	13
2.2.3 Schedule feasibility.....	14

3.3	Data collection.....	14
3.4	Methodology process	15
3.5	Requirements.....	26
3.5.1	Types of Requirement.....	26
CHAPTER 4	28
4.1	Overview	29
4.2	Context diagram-0.....	29
4.3	Data flow Diagram-1.....	30
4.4	Use Case Diagram.....	31
4.5	Sequence diagram	32
	• Sequence diagram for Course withdrawal process :	32
	• Sequence diagram for User login :	33
4.6	ER Diagram.....	34
4.7	Relational Model	35
CHAPTER 5	36
5.1	Overview	37
5.2	Testing methodologies	37
5.2.1	Requirements Testing	37
5.2.2	Unit Testing Results.....	41
5.2.3	System Testing Results	44
5.2.4	Acceptance System Results	44
5.3	Discussion and evaluation	45
CHAPTER 6	46
6.1	Overview	47
6.2	Summary of the project.....	47
6.3	Achieved objectives	47
1.	Test Design Specification:.....	49
2.	User Acceptance:	50
6.4	Main contributions of the work.....	50
6.5	Limitation	50
6.6	Future Work	51

Appendices.....	53
-----------------	----

List of Tables

Table 1: Illustrates the comparison of the trsditional system and the automated system	10
Table 2: Illustrates the comparison of the withdrawal system in several universities	11
Table 3: Technical Feasibility Study	13
Table 4: Operational Feasibility.....	14
Table 5: Schedule Feasibility.....	14
Table 6: Scrum meetings	17
Table 7: S1 Sprint Planning	18
Table 8: S1 Development.....	18
Table 9: S1 Sprint Review	18
Table 10: S2 Sprint Planning	19
Table 11: S2 Development.....	20
Table 12: S2 Sprint Review	20
Table 13: S3 Sprint Planning	21
Table 14: S3 Development.....	21
Table 15: S3 Sprint Review	21
Table 16: S4 Sprint Planning	22
Table 17: S4 Development.....	23
Table 18: S4 Sprint Review	23
Table 19: S5 Sprint Planning	24
Table 20: S5 Development.....	25
Table 21: S5 Sprint Review	25
Table 22: Functional requirements	26
Table 23: Non_ Functional requirements	27
Table 24: Requirements Testing	37

List of Figures

Figure 1: Scrum process.....	4
Figure 2: Gantt chart	5
Figure 3: Scrum framework	16
Figure 4: Context diagram-0	29
Figure 5: Data flow Diagram-1	30
Figure 6: Use Case Diagram	31
Figure 7 Sequence diagram for Course withdrawal process.....	32
Figure 8 Sequence diagram for User login	33
Figure 9: ER Diagram.....	34
Figure 10: Relational Diagram.....	35

List of abbreviations

WISE	the World Islamic Sciences and Education university
PHP	Hypertext Preprocessor
IT	Information Technology
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
Ajax	Asynchronous JavaScript and XML
S	Sprint
UI	User Interface
ER	Entity Relationship
DB	Database

CHAPTER 1

INTRODUCTION

1.1 Overview

This project introduces an electronic service for course withdrawals in the World Islamic Sciences & Education University (WISE), this service will be considered as a subsystem or component in the Academic services portal, this service, eliminating the need for cumbersome paper processes in educational institutions with online procedures and tracking the request's status, with an automated real-time notification.

It automates course withdrawals through a web-based system using the PHP programming language as server-side code with the aid of HTML, CSS, JavaScript, and Ajax techniques. In addition to that, it connects with the student information database, which can be sent to the browser for rendering, allowing students to request withdrawals and administrators to process and approve or reject them efficiently.

1.2 Problem Statement

The Admission and Registration department at WISE University plays an important role, it provides many services, and one of these services is course withdrawal. However, completing this task requires a lot of time and effort from the student. It involves obtaining a confirmation paper from the admission and registration department, then approval from the respective course professor, and the head of the student's department. This document should be approved by the faculty dean and finally, by the Admission and Registration department to complete the request by updating the student registration records.

This process not only consumes time but also entails a risk of compromising student's privacy. In addition, the relevant individuals including the respective course professor, the heads of departments, faculty's' deans, and staff members of the

Admission and Registration department need to be available in their offices to provide timely confirmation for the students who are willing to withdraw from the course.

Furthermore, this process may contradict other processes such as the deprivation of the student. Hence, this process and other processes were limited by the university's calendar. Since the legacy process is mainly based on papers and users, the probability of human error is increased as a chance of missing the document or being damaged.

1.3 Project objectives

1. Automation of the legacy withdrawal process.
2. Validating and controlling contradictions processes such as withdrawal.
3. Reducing human errors.
4. Reducing the time and effort taken to complete the procedures.
5. Protecting student's information privacy.
6. Making the process easier and more comfortable for the students and users involved in this procedure.
7. Eliminating the necessity of timely confirmation from the relevant individuals.

1.4 Research Strategy

The Agile methodology is the most suitable approach for developing a course withdrawal system due to its flexibility, iterative nature, and focus on collaboration. "Agile allows the system to develop as a series of versions or increments with stakeholders involved in version specification and evaluation, with frequent delivery of new versions for evaluation. Additionally, it focuses on the working code".¹

Agile's iterative development process enables us to deliver value early and often, making it well-suited for projects with changing priorities and requirements. Based on that we decided to make our project Agile.

¹ Kenneth Kendall & Julie Kendall, Systems Analysis and Design 9e, Prentice Hall.

Nowadays different frameworks use agile principles and provide guidance on how to implement Agile in practice. The most popular agile frameworks include Scrum, Kanban, Crystal family, and XP.

Scrum is one of the most popular ways to implement Agile. In Scrum, iterations are called sprints, and they usually last two weeks. Scrum, like most Agile methods, favors cross-functional teams.

Scrum framework has a flexible and adaptable nature, and it is well-suited for small teams because it is designed to encourage collaboration and communication among team members and it provides transparency through regular in-sections and adaptability, ensuring consistent value delivery throughout the project. Therefore, Scrum is a suitable methodology for our project purposes, and we can efficiently develop and deliver the course withdrawal system while adapting to any changes along the way.

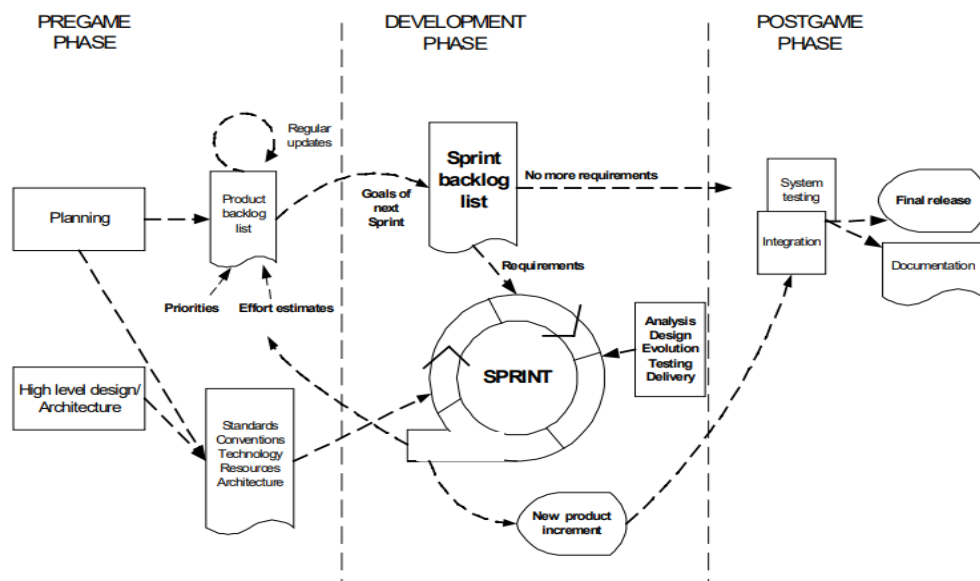


Figure 1: Scrum process²

² [pekka abrahamsson, outi salo, jussi ronkainen & juhani warsta agile software development methods](#)

1.5 Scope (boundary)

In our project, the scope will be limited to the withdrawal procedure, we will focus on automation of the legacy paper-based withdrawal process and make it a web-based service as a component of the Academic services portals for the students and the other corresponding users. Mainly this subsystem will allow students with various academic degrees (diploma, bachelor, and higher studies) to request course withdrawal concerning the university's regulations and policies. In addition, allowing the systems' users to track the status of the submitted requests with real-time automated notifications.

1.6 Gantt chart

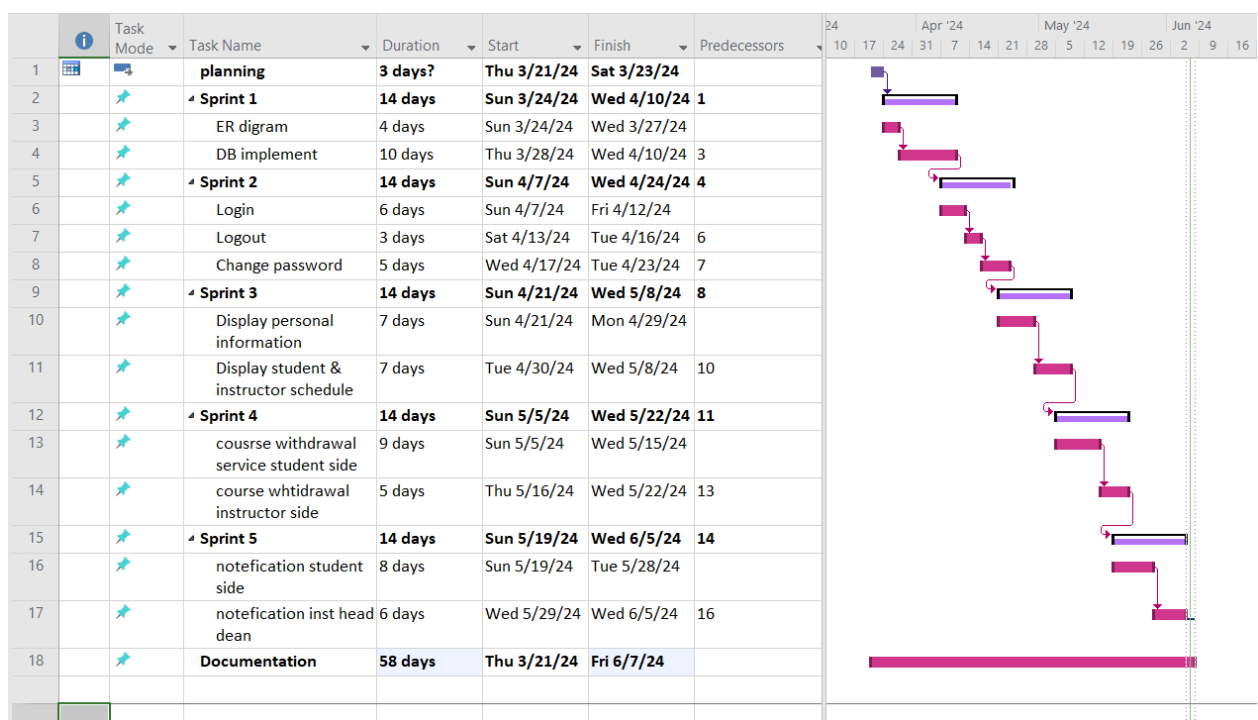


Figure 2: Gantt chart

1.6 Project outline

- Chapter 2 presents related works and a literature review that demonstrates the system's efficiency.
- Chapter 3 discusses the project's approach, functional and non-functional needs, and the feasibility assessment.
- Chapter 4 explains the system's functionality using a series of graphs for clarity.
- Chapter 5 discusses implementation and assessment, summarizing each application and the steps taken in execution.
- Chapter 6 summarizes future research for the project..

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

Several universities have changed their withdrawal system processes to be electronic; others, such as The World Islamic and Science Education University, continue to do the withdrawal process manually. This chapter includes a comparison of various withdrawal systems implemented by other Jordanian educational institutions and sheds light on some of the issues and challenges that were addressed by students through conducted interviews.

2.2 Related Work

2.2.1 Withdrawal system automation

Many Jordanian universities have started to automate their systems; to enhance efficiency, accuracy, and reduce processing time. In addition to that, the online system is available 24/7 allowing students to use the system at their convenience, beyond the typical office hours and working hours.

From the perspective of the student, there is a strong need to utilize modern technology to make many traditional university services such as the course withdrawal service more accessible. However, some of the Jordanian universities, like Al-Balqa' Applied University, Princess Sumaya University, The Hashemite University, and Al Hussein Technical University, have implemented this system, while others, such as the WISE University and Al-Isra University, continue to rely on a manual process for course withdrawal service. It's important to note that each university operates under its own set of regulations, which vary the comparison criteria between them.

2.2.2 Challenges in the traditional system

An office visit, paperwork completion, and manual process completion are all parts of the traditional course withdrawal service that are challenging and uncertain. It consumes time and makes you unsure about its status. There's always the possibility of misplacing requests or entering wrong data because of human errors.

Overall, the automated system is much better. It's faster, more convenient, and less prone to mistakes.

2.2.3 Current service vs. Automated service

The electronic system represents a strategic approach to enhance business processes for organizations, by minimizing human intervention and errors. Within our project, the automated course withdrawal process enhances accessibility, service delivery, and quality, as well as its cost-effectiveness and minimizes inconvenience. On the other hand, the Manual service may offer flexibility but is often slower, prone to errors and it requires greater expenses over the long run.

Here are some differences that will be explained: -

- **Efficiency:** - The automated withdrawal service serves to streamline the entire withdrawal services, saving time and effort for both students and administrators alike, by eliminating the need for manual paperwork or in-person visits.
- **Accessibility:** - In the automated course withdrawal service, the students can initiate withdrawal requests conveniently through a web-based system from anywhere with internet access, without the need to visit administrative offices providing greater convenience and accessibility, while the traditional system operates within specified office hours, restricting students' ability to submit withdrawal requests beyond of these hours.
- **Processing Time:** - Using the automated course withdrawal service, the requests are processed more quickly as they are submitted electronically, which reduces the time that's required for the process to be completed. In the traditional system, manual processing of paper forms can lead to delays in course withdrawal requests, especially during peak times.

- **Accuracy:** - The automated withdrawal service reduces the possibility of human errors in processing withdrawal requests, ensuring that requests are handled accurately and promptly, and the request's owner can keep track of the status of the request.
- **Inconvenience:** - Students must visit administrative offices in person to submit withdrawal requests, and visit different buildings on campus, which can be time-consuming and inconvenient, especially during peak times. And that doesn't exist in the electronic system.
- **Cost:** - Using the automated course withdrawal service involves initial costs for the system development, but saves money, in the long term. By reducing manual processing, it reduces administrative costs (ex: papers) and minimizes the risk of errors. This saves money by avoiding issues that appear while handling withdrawal requests.

Table 1: Illustrates the comparison of the traditional system and the automated system

	Traditional system	Automated system
Efficiency	No	Yes
Availability	University office hours	Anytime
Processing Time/speed	Long Slow	Short Fast
Accuracy	Low	High
Inconvenience	Yes	No
Cost	High (in the long term)	Low (in the long term)
Track Status	At final stage	In each change in the request

These withdrawal systems differ in many features, so we interviewed some students from different universities to compare their experiences. The following table is a summary of our findings.

Table 2: Illustrates the comparison of the withdrawal system in several universities

	Al-Balqa' Applied University	Princess Sumaya University	Al-Hussein Technical University	The Hashemite University
Tracking	NO	NO	No	NO
The ability to cancel the request	NO	Yes	No	NO
Notification	NO	Yes	Yes	NO

2.3 Summary

This chapter validates the challenges of using the traditional withdrawal system and sheds light on the issues that students may encounter, which makes it inefficient and poses risks. In addition to that, this chapter highlighted some of the existing automated withdrawal systems that have been already implemented in other Jordanian institutes, it demonstrates how these systems provide enhanced solutions for the problems noted.

CHAPTER 3

METHODOLOGY

3.1. Overview

This chapter includes the definition of the feasibility study, the tools employed in our work, the gathered requirements (both functional and non-functional), and the methodological process used for the project's implementation.

3.2. Feasibility Study

The feasibility study is a way to evaluate whether or not a project plan could be successful. A feasibility study evaluates the practicality of your project plan to judge whether you're able to move forward with the project.¹ It includes three types: operational, technical, and economic feasibility.

3.2.1 Technical feasibility

A technical feasibility study reviews the technical resources available for the project, which determines if the right equipment, enough equipment, and the right technical knowledge are provided to complete this project's objectives.

Table 3: Technical Feasibility Study

Technical aspects	Financial costs
Development costs	1000\$
Testing costs	500\$
Marketing costs	200\$
Security and backup costs	100\$
Total cost	1800\$

3.2.2 Operational feasibility

An operational feasibility study evaluates whether the organization can complete this project or not. This includes staffing requirements, organizational structure, and any applicable legal requirements. At the end of the operational feasibility study, your team will have a sense of whether you have the resources, skills, and competencies to complete this work.

Table 4: Operational Feasibility

Process	Percentage
Readiness and Training	90%
Maintenance Viability	75%
Workflow Efficiency	75%
Performance	85%
Process Integration	80%

2.2.3 Schedule feasibility

Schedule feasibility assesses how likely a project will be completed within its proposed timeframe. This is a crucial feasibility study because the results will determine whether your project will be successful. The study identifies key constraints on the project that can affect the timeline, including internal and external constraints like regulations, politics, budgets, and technology. ³

Table 5: Schedule Feasibility

Phase	Estimated time (in weeks)
Estimate project completion time	11
Analysis of Time Risks	2
Scheduling Activities	1
Estimated Project Delay	2

3.3 Data collection

Gathering requirements for a project or data collection refers to the process of gathering information about the project and the stakeholders' perspectives, which helps in decision-making and project planning. Sources of this data range from stakeholders' interviews, surveys, existing customer data, analytics data, and market research. The objective of this process is to provide a complete understanding of the project requirements, goals, and constraints, which can be used to guide project planning and decision-making.

Interviews are one of the most efficient and popular methods that are used to gather data and requirements, it allows you to communicate directly with the stakeholder. This can

³ <https://lucidspark.com/blog/conduct-a-feasibility-study>

provide an opportunity for the stakeholders to share their ideas and concerns to be discussed. Interviews can be conducted in person, over the phone, or via video conference, they can be categorized into structured or unstructured depending on the goals of the interview and the preferences of the stakeholders.

Therefore, we have chosen the interview method for gathering data and formulating requirements. As we gathered data, we navigated through three distinct stages, the first stage involves understanding the problem statement, which involved conducting interviews that are highlighted in Chapter 2. Following this, we delved into understanding the constraints and regulations specific to WISE University. Finally, we engaged in interviews to shape our final requirements.

3.4 Methodology process

A methodology refers to a specific set of procedures with a defined set of rules, methods, tests, activities, deliverables, and processes that are typically designed to address a particular problem. It provides a structured approach to problem-solving by outlining a series of steps and activities that should be followed to achieve the desired outcome.

Scrum is one of the most famous development methodologies. It involves several environmental and technical variables (e.g. requirements, resources) that are likely to change during the process, which affects the development process in a way that makes it unpredictable and complex, if not handled properly. This requires flexibility in the system's development process to be able to respond to the changes. In addition to that, Scrum helps to improve the existing engineering practices in an organization, for it involves frequent management activities aiming at consistently identifying any deficiencies or impediments in the development process as well as the practices that are used.

The main roles in scrum:-

- Scrum master.
- Product owner.
- Scrum team.
- Customer.
- Management.

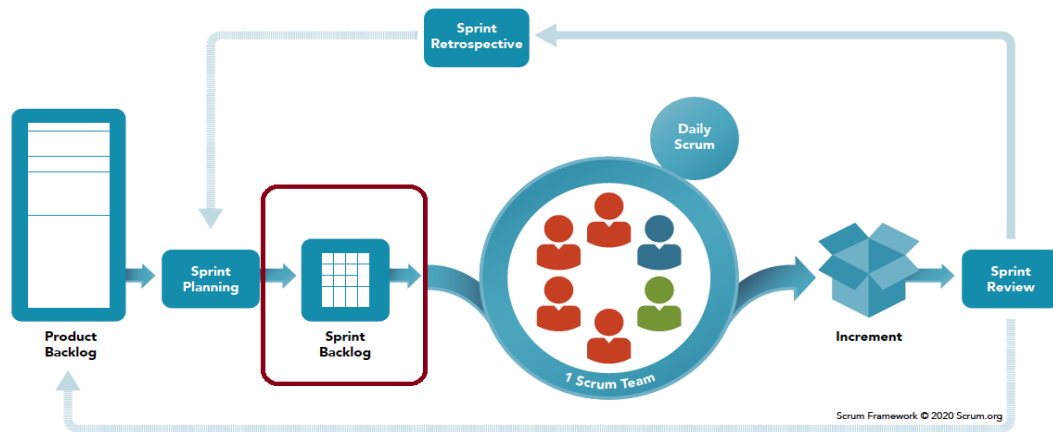


Figure 3: Scrum framework⁴

Sprints are iterative cycles where the functionality is developed or enhanced to produce new increments. Each Sprint includes the traditional phases of software development: requirements, analysis, design, evolution, and delivery phases. The architecture and the design of the system evolved during the Sprint development. One Sprint is planned to last from one week to one month. Also, there may be more than one team building the increment. [2]

Scrum likes to call the output of a sprint the "Potentially Shippable Product".

Backlog:

- **Product Backlog:** As described in the Scrum Guide⁵, "Is an emergent, ordered list of what is needed to improve the product. It is the single source of work undertaken by the Scrum Team"⁶ and it is the to-do list for the project. It contains a list of all the features the Product Owner would one day like to see in their software. It's prioritized, estimated, and what the team draws from when deciding what to do in an iteration or sprint.

⁴ scrum.org, <https://www.scrum.org/resources/blog/what-sprint-backlog-and-how-it-connects-other-elements-scrum> (accessed May 27, 2024).

- **Sprint Backlog:** What the team commits to each sprint is tracked in a Sprint Backlog. It contains a list of user stories the Product Owner would like to see developed for that sprint.

While the Product Backlog identifies ALL of the pending features of the product being developed, the Sprint Backlog identifies only that smaller list of Product Backlog items that have been jointly agreed between the product owner and the team as goals to be accomplished during the current sprint.

Table 6: Scrum meetings

Meeting Name	Frequency	Length	Participants
• Grooming	Once per sprint, prior to the start of the sprint	4 hours	PO, Leads
• Sprint planning	Once per sprint (day 1)	4 hours	PO, Leads
• Daily Scrum	Daily (except the first day)	15 minutes	Full team
• Sprint review ("Demo")	Once per sprint (last day of sprint)	2 hours	PO, Leads
• Sprint Retrospective	Once per sprint, after the sprint closes	1.5 hours or less	Full Team

- **Backlog Grooming Meeting:** the time in this meeting will be used to help create or review emergent product backlog items, as well as progressively refine large items into smaller items.
- **Sprint Planning Meeting:** In this meeting, the development team collaborates with the Product Owner to review and prioritize user stories. The team then estimates the effort required for each story and selects those that can be completed within the duration of the sprint.
- **Daily Scrum:** The team uses this time to report any issues or progress on their tasks using the format, 1) What I did yesterday 2) What I plan to do today 3) Call out any blockers. Normally the Product Owner is present during all daily scrum meetings to assist in any way.
- **Sprint Review Meeting** ("Demo"): A crucial event at the sprint's end, showcasing completed work, reviewing the product, and suggesting changes or improvements, which will go into the backlog.

- **Sprint Retrospective Meeting:** This meeting is held to facilitate a team's reflection on **their progress**.

- **Sprint 1 :**

1. **Sprint Planning**

Table 7: S1 Sprint Planning

ID	User Story	Description	Days
S101	As a user, I want all of my information to be stored in a centralized database so that I can retrieve them and modify them at any time.	Design and implement a database comprising all the necessary tables to store and manage users' information. Ensure the database is secure, and scalable and supports seamless retrieval and modification of data.	14

2. **Development**

Table 8: S1 Development

ID	Tasks
\$101	- designing an ERD diagram for the database - designing a Relational diagram - implement the database - handling constraints and errors

3. **Sprint Review**

Table 9: S1 Sprint Review

ID	Tasks
\$101	- designing an ERD diagram for the database - designing a Relational diagram - implement the database - handling constraints and errors

- **Sprint 2 :**

1. Sprint Planning

Table 10: S2 Sprint Planning

ID	User Story	Description	Days
S201	As a student, I want to log in to the course withdrawal system, So that I can securely access my account, view my schedule, and my information, and submit a course withdrawal request.	Implement a login functionality for registered students.	3
S202	As an instructor, I want to log in to the course withdrawal system, So that I can securely access my account, view my schedule, and my information, and approve or reject a student request.	Implement a login functionality for the instructor.	3
S203	As a head of department or dean, I want to log in to the course withdrawal system, So that I can securely access my account, view my information, and approve or reject a student request.	Implement a login functionality for the head of department and dean.	3
S204	As a user, I want to log out of the course withdrawal system, so that I can securely end my session and protect my account from unauthorized access.	Implement a logout functionality for the users.	2
S205	As a user, I want to change my password, So that I can enhance the security of my account and ensure only I have access to it.	Implement a change password functionality for all users to ensure that they can securely update their passwords.	3

2. Development

Table 11: S2 Development

ID	Tasks
S201, S202, S203	- Create login form UI. - Implement backend authentication. - Handle authentication errors.
S204	-Implement a logout button. - Clear session data. - Redirect to the login page.
S205	-Create a change password UI. -Implement change password function &update the database. - Validated new passwords at the security policies and provided feedback on success or failure.

3. Sprint Review

Table 12: S2 Sprint Review

ID	Completed Tasks	Acceptance Criteria Met?	Feedback / Next Steps
S201, S202, S103	- Create login form UI. -Implement backend authentication for each role.- Handle authentication errors.	Every user can access the system based on their role. Met: Yes.	Users can successfully log in.
S204	-implemented the logout button functionality. - Cleared session data securely to ensure user privacy. -Redirect users to the login page after logout for improved security.	Yes	Logging out works as expected.
S205	-Created the UI for the change password function. -Implemented backend logic for securely changing passwords. -Validated new passwords against security policies and provided feedback on success or failure.	Every user can change their password successfully. Met: Yes.	Students appreciated this function as very useful.

- **Sprint 3 :**

1. **Sprint Planning**

Table 13: S3 Sprint Planning

ID	User Story	Description	Days
S301	As a user, I want to view my personal information, So that I can verify my personal details.	Implement a view personal information functionality for all users to ensure that they can access and verify their personal details.	8
S302	As a student or instructor, I want to view my schedule, So that I can see the courses I am enrolled in and plan my time accordingly.	Implement the functionality of a view registered course for the student and instructor to display and verify their courses.	6

2. Development

Table 14: S3 Development

ID	Tasks
S301	-Create User Information UI. -Implement a database query to fetch user information based on user ID. - Handle authentication errors.
S302	-Create a UI design for the registered courses. -Implement the backend logic to retrieve the student's registered courses from the database. -Handle any potential errors, such as data retrieval failures.

3. Sprint Review

Table 15: S3 Sprint Review

ID	Completed Tasks	Acceptance Criteria Met?	Feedback / Next Steps
S301	-Created the UI for displaying user personal information. -Implemented backend logic to fetch user information from the database. -Displayed the fetched user information on the UI.	The user can navigate to the Personal Information page and see their personal details. Met: Yes	Users found the interface easy to navigate and very clear.

S302	-Created the UI for displaying the student schedule. -Implemented backend logic to retrieve registered courses from the database.	- The user can navigate to the "Schedule" page and see their enrolled courses. Met: Yes - The schedule information is accurate and reflects recent changes. Met: Yes	Users found the interface easy, clear, and reflects recent changes.
-------------	--	---	---

- **Sprint 4 :**

1. **Sprint Planning**

Table 16: S4 Sprint Planning

ID	User Story	Description	Days
S401	As a student, I want to choose a course from my schedule to withdraw from, So that I can submit a withdrawal request.	Implement a withdrawal request from the student side.	4
S402	As a student, I want to undo my withdrawal request, So that I can cancel it if I change my mind.	Provide the student with the option to undo the course withdrawal.	2
S403	As a student, I want to track the status of my withdrawal request, So that I can see if it has been approved or rejected.	Implement a course status tracking (request status) functionality for students.	3
S404	As an instructor, head of department, or dean, I want to approve or reject withdrawal requests So that I can manage course enrollments.	Implement a request processing functionality.	5

2. Development

Table 17: S4 Development

ID	Tasks
S401	<ul style="list-style-type: none"> - Add a withdrawal option to each course in the schedule UI when the student clicks the withdrawal request in the sidebar. - Implement the backend to update the student's course status and send the request to the instructor. - Validate and submit the withdrawal request to the backend. - Handle any potential errors.
S402	<ul style="list-style-type: none"> - Implement an option to undo a withdrawal request. - Validate and process the undo request in the backend. - Update the UI and handle errors.
S403	<ul style="list-style-type: none"> - Display the status of each withdrawal request on the schedule UI. - Fetch the status from the backend and update it in real-time. - Handle any potential errors.
S404	<ul style="list-style-type: none"> - Design the UI for an instructor, department head, and dean to display withdrawal requests. - Implement the backend logic to handle approval/rejection by the instructor, head of department, and dean. - Update the database based on the request status. - Handle any potential errors.

3. Sprint Review

Table 18: S4 Sprint Review

ID	Completed Tasks	Acceptance Criteria Met?	Feedback / Next Steps
S401	<ul style="list-style-type: none"> - Added withdrawal option to the schedule UI when the student clicks withdrawal request in the sidebar. - Implemented backend functionality to update course status and send withdrawal requests to instructors. - Withdrawal requests can now be submitted securely and efficiently. 	<p>Students can view their schedule and initiate a withdrawal request for any course. Met: Yes.</p> <p>- The student expected to receive a notification about the status of the request, not only on the schedule. Met: No.</p>	<ul style="list-style-type: none"> - Users appreciated the intuitive UI design and ease of navigation. - Implementing a notification feature for the students.

S402	-Implemented functionality for students to undo withdrawal requests. -Ensured that undo requests are processed accurately.	Students now have the flexibility to change their minds about withdrawal decisions. Met: Yes.	Students appreciated this function as very useful
S403	-Implemented real-time status tracking for withdrawal requests.	Student can stay informed about any updates in their request state. Students can track the status of their withdrawal request in real-time. Met: Yes.	Student feedback was positive about this feature
S404	-Design the UI for the instructor, department head, and dean to display withdrawal requests. -Implement the backend logic to handle approval/rejection by the instructor, head of department, and dean. -Update the database based on the request status.	-Instructor, department head, and dean can view and approve/reject withdrawal requests. Met: Yes.	The Instructor, department head, and dean suggested receiving new notifications when a new student requests for course withdrawal.

- **Sprint 5 :**

1. **Sprint Planning**

Table 19: S5 Sprint Planning

ID	User Story	Description	Days
S501	As a student, I want to receive real-time notifications about the status of my withdrawal request, So that I am always informed about its progress.	Implement a notification function that informs students about the status of their course withdrawal requests.	6
S502	As an instructor/department head/dean, I want to receive notifications if any new withdrawal requests, So that I can take timely action.	Implement a notification function that informs the instructor, department head, and dean of any new withdrawal requests.	8

2. Development

Table 20: S5 Development

ID	Tasks
S501	<ul style="list-style-type: none"> -Design the notification function architecture. -Implement backend logic to send notifications when a withdrawal request status changes. -Create the frontend component to display notifications to students. -Handle any potential errors.
S502	<ul style="list-style-type: none"> - Implement backend logic to send notifications to the instructor, head of department, and dean if any new withdrawal requests. -Update the withdrawal request function by adding the notification function to it. -Handle errors and delivery of notifications.

3. Sprint Review

Table 21: S5 Sprint Review

ID	Completed Tasks	Acceptance Criteria Met?	Feedback / Next Steps
S501	<ul style="list-style-type: none"> -Designed the notification function architecture. -Implemented backend logic to send notifications when a withdrawal request status changes. -Created the frontend component to display notifications to students. 	<ul style="list-style-type: none"> -Students receive a notification directly when the status of their withdrawal request changes. Met: Yes. <ul style="list-style-type: none"> -Notifications include clear information about the status ex. (approved or rejected by the instructor). Met: Yes.	Students appreciated this function as very useful.
S502	<ul style="list-style-type: none"> -Implemented backend logic to send notifications to the instructor, head of department, and dean if any new withdrawal requests. -Updated the withdrawal request function by adding the notification function to it. -Handled errors and ensured reliable delivery of notifications. 	<ul style="list-style-type: none"> - Instructors, department heads, and deans receive notifications as soon as a withdrawal request is submitted. Met: Yes. <ul style="list-style-type: none"> -Notifications clearly indicate that action is required and provide necessary details about the request. Met: Yes.	- Instructors, department heads, and dean's feedback was positive about this feature.

3.5 Requirements

The requirements for a system are the descriptions of what the system should do—the services that it provides and the constraints on its operation. These requirements reflect the needs of customers for a system that serves a certain purpose such as controlling a device, placing an order, or finding information.⁵

3.5.1 Types of Requirement

Software system requirements are often classified as functional requirements or non-functional requirements:

- **Functional requirements** These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations. In some cases, the functional requirements may also explicitly state what the system should not do. [5]

Table 22: Functional requirements

No.	Requirement	Description
1	Login	The users can log in to the system.
2	Logout	The users can log out of the system.
3	Change password	Users can change their password
4	View personal information	All the users can display their personal information.
5	View student schedule	The student can view all the courses he has registered for
6	View instructor schedule	The instructor can view all the courses he has registered for.
7	Request for course withdrawal	Students must be able to select a course and submit a withdrawal request.
8	Undo Withdrawal Request	Students can undo the withdrawal request before it is reviewed by the instructor.
9	Track withdrawal request status	Students must be able to view the status of their withdrawal requests in real-time.
10	Approve/ Reject the course withdrawal request	Instructor, head of a department, and dean can view, approve, or reject withdrawal requests.
11	Notifications	The system must send notifications to students about the status of their withdrawal requests. and if any new request to the instructor, head of department, and dean.

⁵ Ian Sommerville ,SOFTWARE ENGINEERING,9e

- **Non-functional requirements** These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and constraints imposed by standards. Non-functional requirements often apply to the system as a whole, rather than individual system features or services. [5]

Table 23: Non_Functional requirements

No.	Requirement	Description
1	Usability	Our system was designed with a focus on simplicity, user-friendliness, and responsiveness.
2	Security	Our project features an authentication system that requires users to log in to access the screens. Users are identified by a combination of a username and a password. Access to the system and the main screen is restricted and granted only upon successful login.
3	Performance	The performance dimension of our projects guarantees a fast and efficient user experience and ensures that the system responds quickly to user input while maintaining minimal response times.
4	Reliability	In our system, the probability that it performs correctly during a specific time duration is very high with real-time data updates.
5	Scalability	Our system is designed to scale and accommodate an increasing number of users and requests.

CHAPTER 4

DESIGN MODELS

4.1 Overview

Recognizing the essential role of design models that software engineering, this chapter presents an overview of our system through various kinds of diagrams, including a Context diagram, Dataflow Diagram, Entity Relation Diagram, Use Case Diagram, Sequence Diagram, and Relational Diagram. These diagrams provide a standardized way to visualize the design of a system, making it easier to understand and communicate about the system's structure, behavior, and the relationships among software components and users within the system.

4.2 Context diagram-0

A context diagram is high-level, an important tool for establishing the boundaries of the system being modeled. It doesn't go into the detailed ins and outs of the system, they map out an⁶ entire system in a way that's simple, clear, and easy to understand.

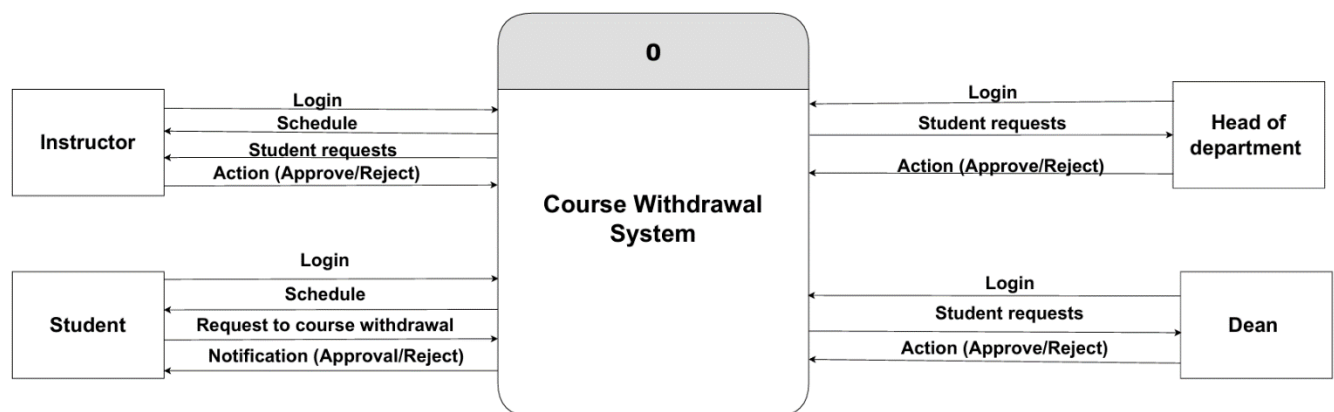


Figure 4: Context diagram-0

⁶ Miro, "what's the context diagram and how do you use it?", <https://miro.com/blog/context-diagram/#:~:text=Context%20diagrams%20are%20high%2Dlevel,system%20and%20each%20external%20element.>, (last access June, 8th 2024)

4.3 Data flow Diagram-1

A data flow diagram (DFD) is a graphical representation of the flow of information through a process or system. It uses standardized symbols and notations to show how data enters a system, is transformed by the system, and exits the system.

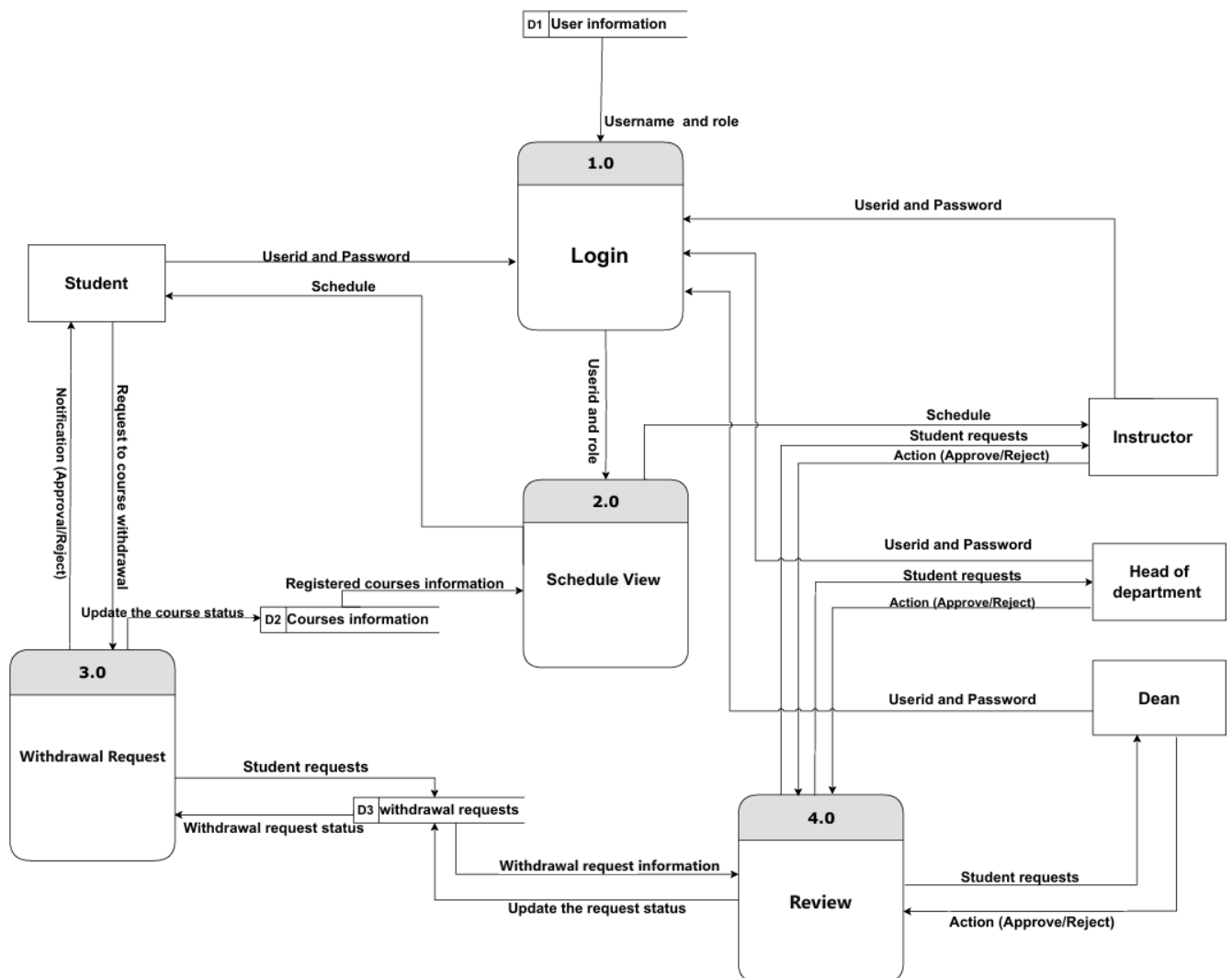


Figure 5: Data flow Diagram-1

4.4 Use Case Diagram

A use-case diagram is one of the UML diagrams that models a system's behavior and helps capture its requirements. Its main purpose is to represent the interactions between the system and its actors.⁷

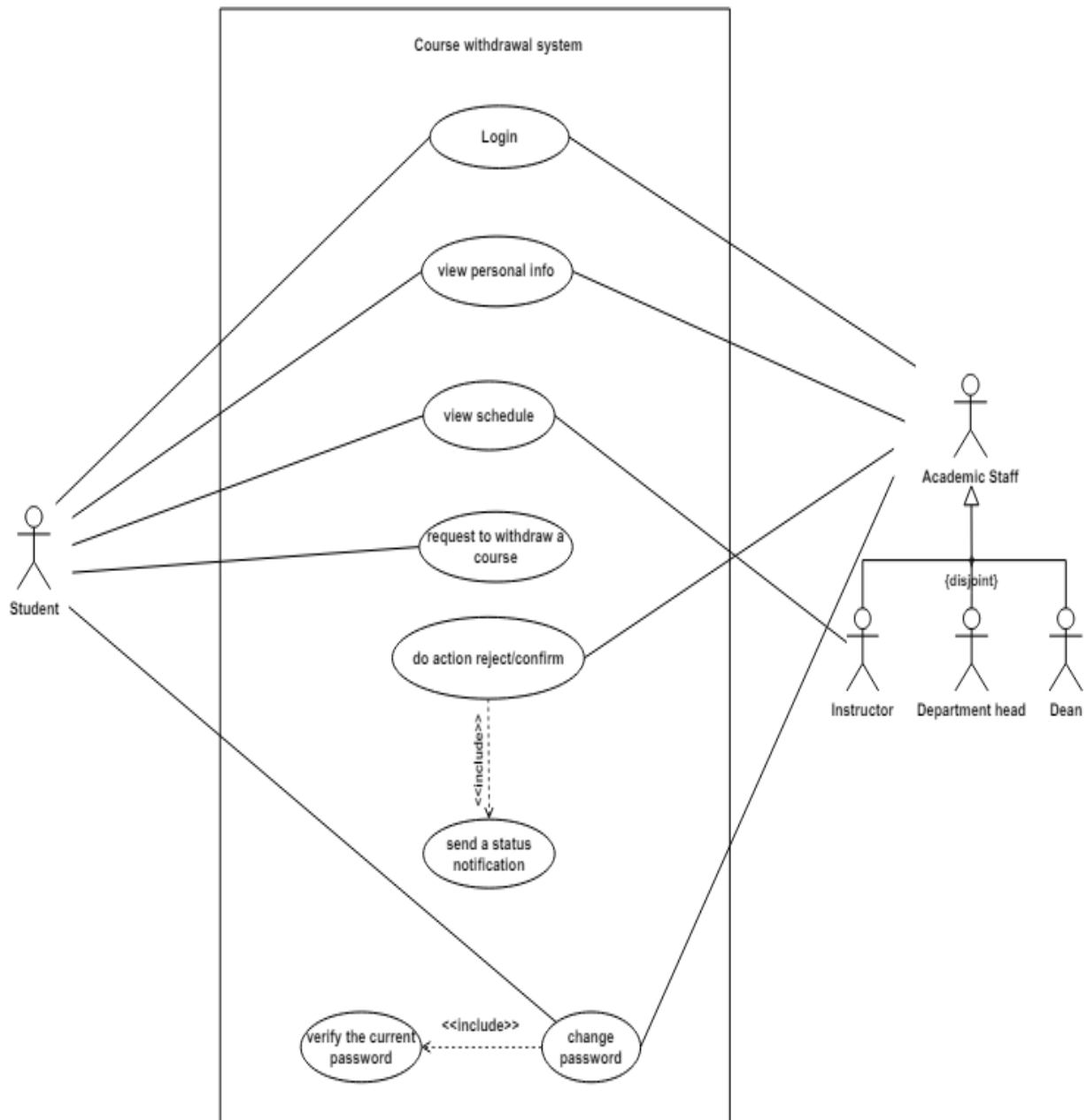


Figure 6: Use Case Diagram

⁷ IBM, "Use-case diagrams in UML modeling", <https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case>, (last accessed, June, 8th 2024)

4.5 Sequence diagram

Sequence diagrams are a popular dynamic modeling solution in UML because they specifically focus on lifelines or the processes and objects that live together, and the messages exchanged between them to perform a function before the lifeline ends.

- Sequence diagram for Course withdrawal process :

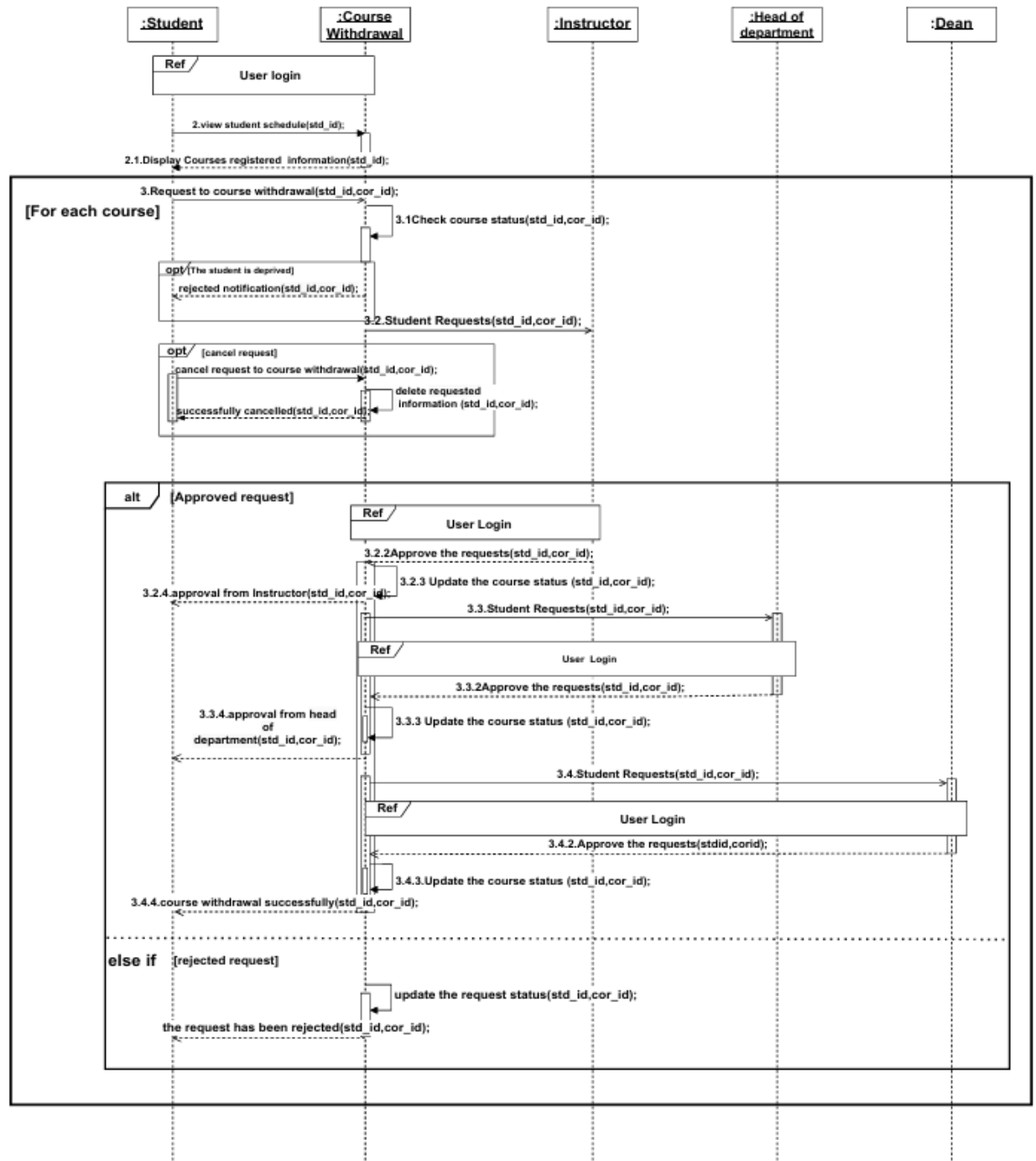


Figure 7 Sequence diagram for Course withdrawal process

- Sequence diagram for User login :

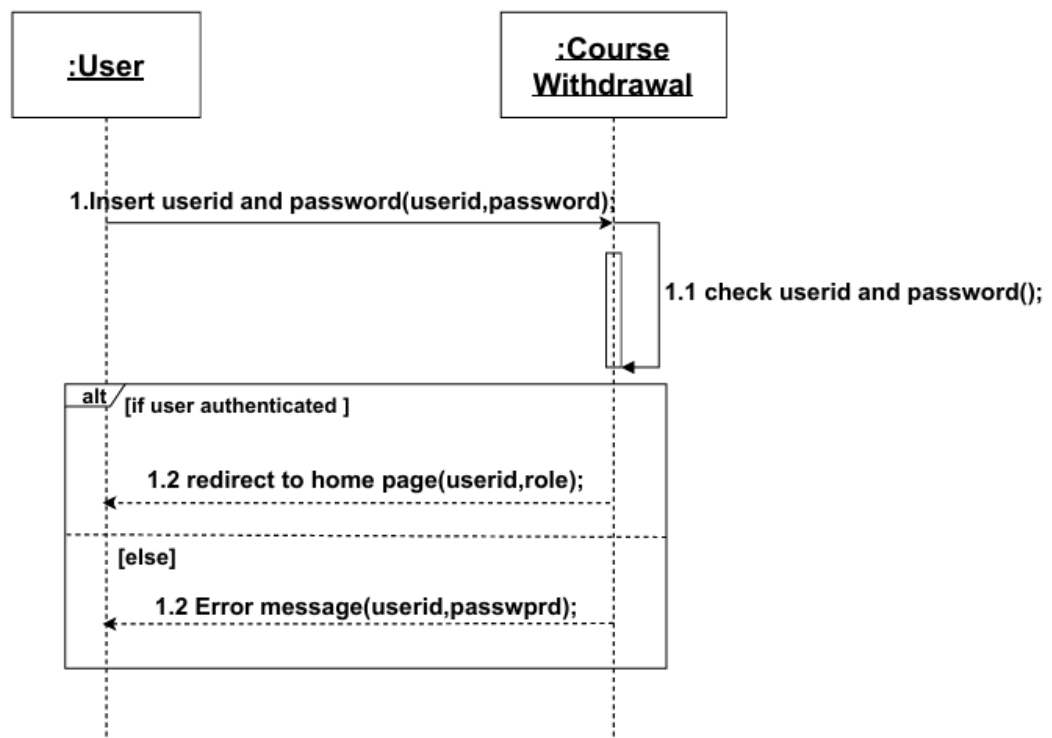


Figure 8 Sequence diagram for User login

4.6 ER Diagram

An Entity-Relationship Diagram (ERD), also known as an ER diagram or ER model, is a graphical representation that depicts the relationships between entities (such as people, objects, places, concepts, or events) within an information system, for database design.

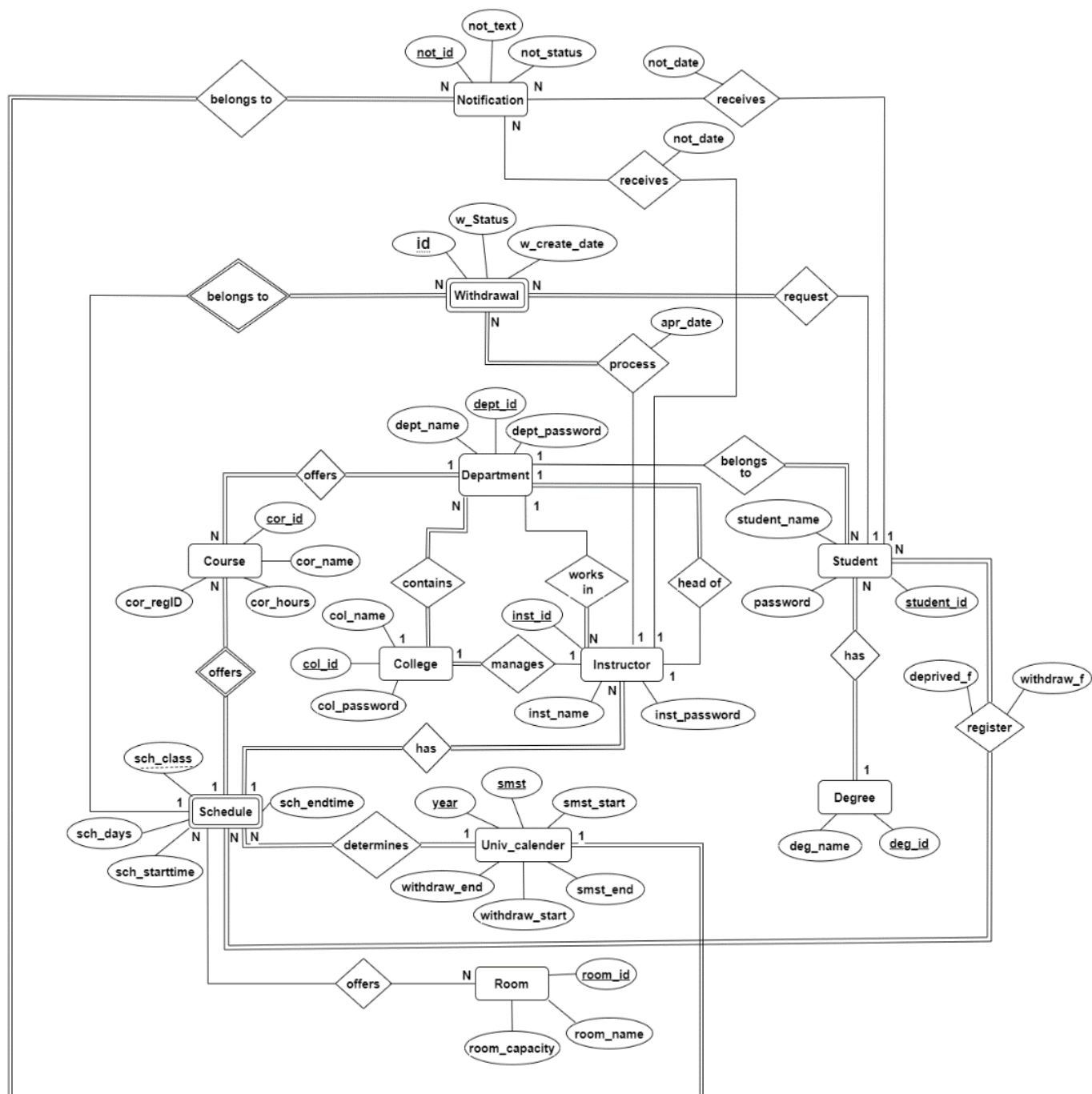


Figure 9: ER Diagram

4.7 Relational Model

Relational diagrams likewise are used to represent relationships between entities in a database, but their use extends beyond database design.

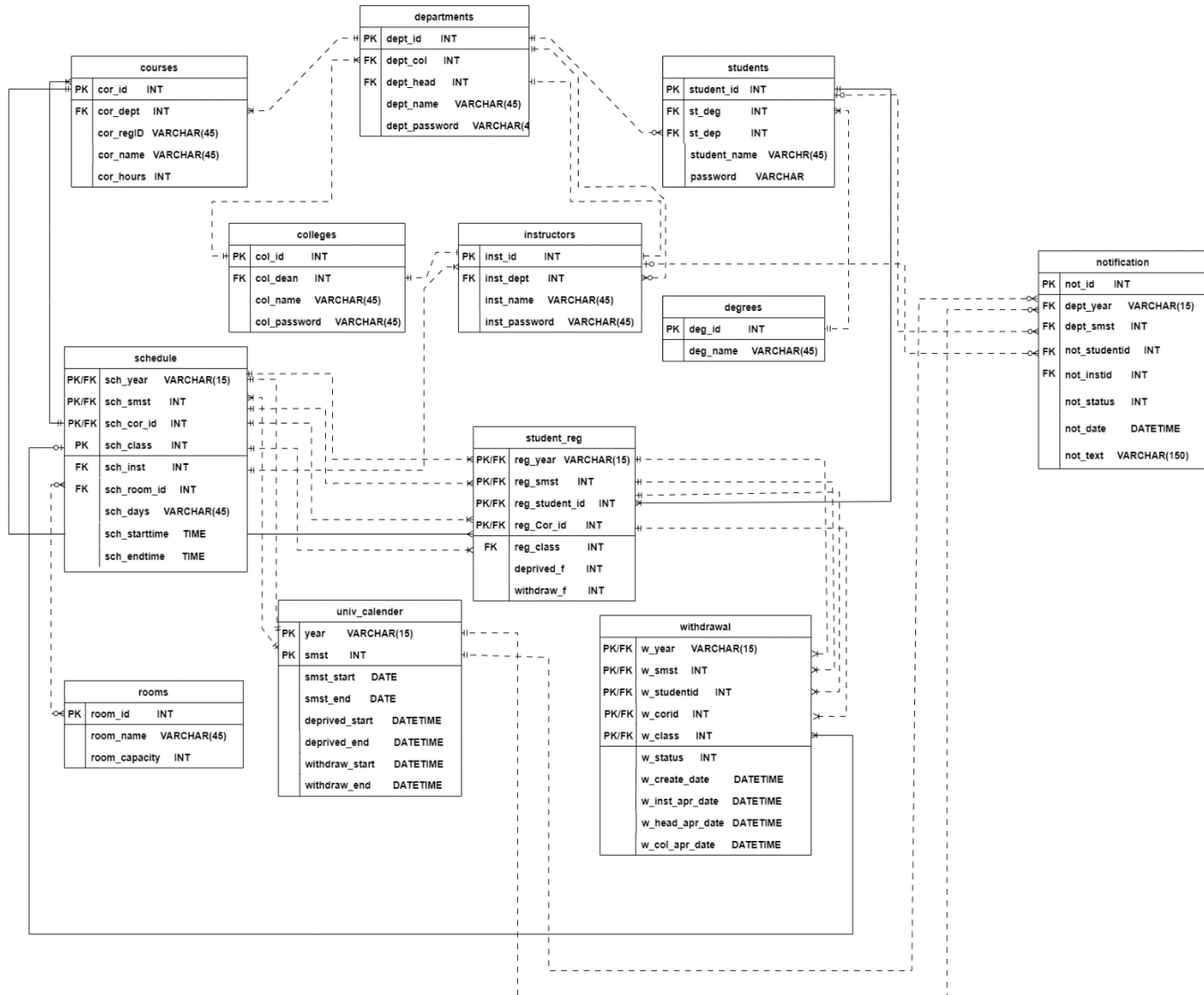


Figure 10: Relational Diagram

CHAPTER 5

EXPERIMENTS AND RESULTS

5.1 Overview

In this chapter, we will explore the importance and outcomes of requirements testing, unit testing, integration testing, and user acceptance testing, highlighting their importance in achieving a robust and reliable software solution. Our focus lies on a meticulous exploration of the system's functionality, ensuring it not only meets but surpasses the predefined criteria.

5.2 Testing methodologies

5.2.1 Requirements Testing

Doing the following test cases for the student:

Student

- Username: 2023100
- Password: 12!##

Instructor

- Username: inst3
- Password: 123

Head of a department

- Username: dept101
- Password: 123

Table 24: Requirements Testing

No.	Test case	Description	Decision	Example
1	Login			
	TC1_01	Student enters a correct Student ID and password.	Pass	Username:2023100 Password: 12!##
	TC1_02	Instructor enters a correct Username and	Pass	Username:inst3 Password: 123

		password.		
	TC1_03	User enters the password only without the username.	Error message “Username is Required!”	Password: 123
	TC1_04	User enters the username only without the password.	Error message “Password is Required!”	Username:inst1
	TC1_04	User enters wrong username or wrong password.	Error message “Wrong username or password.”	Username: 2023101 Password: 1241
2	Change Password			
	TC2_01	User enters their correct current password and two identical new passwords.	Pass	Current password :12!## New password:123 Confirm new password:123
	TC2_02	User enters an incorrect current password and two identical new passwords.	Error message “كلمة السر الحالية غير صحيحة حصل خطأ ما!! ”	Current password :711 New password :275 Confirm new password :275
	TC2_03	User misses a field.	Error message “يجب كتابة كلمة السر في جميع الحقول! حصل خطأ ما!! ”	Current password:123 New password : Confirm new password :
	TC2_04	User enters their correct current password, but not two identical new passwords.	Error message “كلمة السر الجديد غير متطابقة مع تأكيد كلمة السر حصل خطأ ما!! ”	Current password:123 New password :1222 Confirm new password :12333

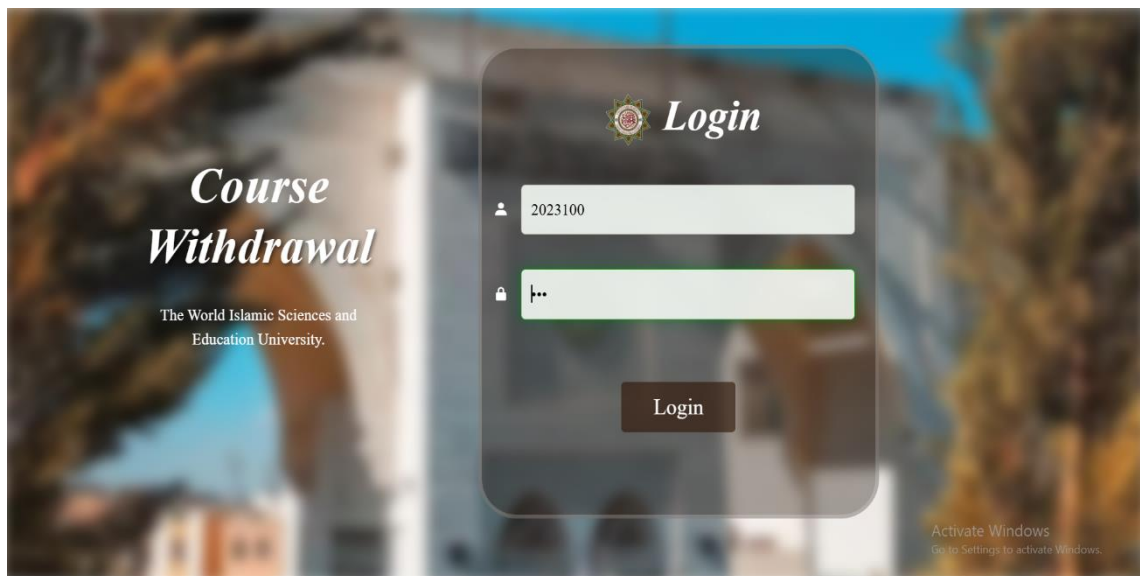
3	Course withdrawal request (From the student side).			
	TC3_01	Student clicks to withdraw from a course that's available in his course	The withdraw button becomes disabled, and the withdrawal status changes to “ بانتظار مدرس المادة ” and then cancel request button appears.	Try to withdraw from the “مادة أساسيات البرمجة” course.
	TC3_02	Student attempts to withdraw from a course that has already been withdrawn	The withdraw button is disabled and the withdrawal status is set to "مسقطه"	Try to withdraw from the “مادة أساسيات البرمجة” course
	TC3_03	The student attempts to withdraw from a course in which he has been deprived.	The withdraw button is disabled and the withdrawal status is set to "محروم"	Try to withdraw from the “مادة أساسيات البرمجة” course
	TC3_04	Student attempts to cancel the withdrawal request before the instructor's approval	pass	Try to cancel the withdrawal request of the “مادة أساسيات البرمجة” course
	TC3_05	Student attempts to cancel the withdrawal request after the instructor's approval	The cancel request button is disabled, and the withdrawal status is changed to "بانتظار رئيس القسم"	Try to cancel the withdrawal request of the “مادة أساسيات البرمجة” course
	TC3_06	Student attempts to withdraw from a course after it has been	The withdrawal button is disabled	Try to withdraw from the “مادة أساسيات البرمجة” course

		rejected		
4	Course withdrawal request (From the instructor's side).			
	TC4_01	The instructor approves the Withdrawal request.	Pass with alert ("هل انت موافق على طلب اسقاط الطالب () لمادة ()؟")	Inst3 approved the withdrawal request for the student 2023100.
	TC4_02	The head of a department rejects the Withdrawal request.	Pass with alert ("هل انت متأكد من رفض طلب اسقاط الطالب () لمادة ()؟")	Dept101 rejected the withdrawal request for the student 2023100.
5	Notifications			
	TC5_01	Student request to course withdrawal.	Pass This request will appear as a new notification in the instructor's withdrawal requests tab.	
	TC5_02	Instructor approves the Withdrawal request.	notification to the student: " قام المدرس بالموافقة على طلب اسقاط المادة أساسيات البرمجة شعبة 1 2024-06-03:50:16 07 جديد "	
	TC5_03	Head of a department rejects the Withdrawal request.	notification to the student: " قام رئيس القسم برفض طلب اسقاط () التاريخ والوقت () الحالة () "	" قام رئيس القسم برفض طلب اسقاط مادة أساسيات البرمجة شعبة 1 2024-06-03:53:10 07-06 جديد "

5.2.2 Unit Testing Results

Unit testing is a type of software testing where individual units or components are tested. The purpose is to validate that each unit of the software code performs as expected. Unit testing is a Whitebox testing technique.

- Login as Student



- Personal information



- request to course withdrawal

localhost:8080 says

هل أنت متأكد من طلب إسقاط المادة: أساسيات البرمجة؟

OK Cancel

الترتيبات على إسقاط المواد

فأمر من الجامعة لتطوف التي لا يربطها الطالب لله اجازت
له التعليمات إجراء تعديل على برنامجهم الدراسي من خلال
مجلس الاسماء الفخرية (مجلس مقرر)

- الاسقاط : هي عملية حذف مادة سبق أن سجلها
الطالب أثناء فترة التسجيل، ولكنه من أجل
تعديل جدولته الدراسية عند الضرورة في آخر
اسابيع من الفصل الدراسي.
- يكون الاسقاط إلكتروني ولا داعي لمراجعة أي
جهة .
- لا تدخل المادة التي يتم اسقاطها بالمعدل.

رقم المادة	اسم المادة	الشعبة	من	الي	الأيام	من	المدرس	القاعة	تقديم طلب اسقاط	حالة الاسقاط
060100	أساسيات البرمجة	1	3	ح	ث	م	د. عبدالله الزقية	مختبر 002 (IT)	طلب الاسقاط	
060101	قواعد البيانات	1	4	ح	ث	م	د. مالك المومني	مختبر 001 (IT)	طلب الاسقاط	
060102	تصميم المنطق الرقمي	3	3	ح	ث	م	د. رجاء مساحدة	102 (IT)	طلب الاسقاط	
060103	بروتوكولات أمن الشبكات	2	3	ح	ث	م	د. مهدي طحيوش	103 (IT)	طلب الاسقاط	
مجموع الساعات : 13							مجموع ساعات الاسقاط : 0			

اهلا بك : يزن أسامة

الترتيبات على إسقاط المواد

فأمر من الجامعة لتطوف التي لا يربطها الطالب لله اجازت
له التعليمات إجراء تعديل على برنامجهم الدراسي من خلال
مجلس الاسماء الفخرية (مجلس مقرر)

- الاسقاط : هي عملية حذف مادة سبق أن سجلها
الطالب أثناء فترة التسجيل، ولكنه من أجل
تعديل جدولته الدراسية عند الضرورة في آخر
اسابيع من الفصل الدراسي.
- يكون الاسقاط إلكتروني ولا داعي لمراجعة أي
جهة .
- لا تدخل المادة التي يتم اسقاطها بالمعدل.

رقم المادة	اسم المادة	الشعبة	من	الي	الأيام	من	المدرس	القاعة	تقديم طلب اسقاط	حالة الاسقاط
060100	أساسيات البرمجة	1	3	ح	ث	م	د. عبدالله الزقية	مختبر 002 (IT)	إلغاء اسقاط المادة	بانتظار مقرر المادة
060101	قواعد البيانات	1	4	ح	ث	م	د. مالك المومني	مختبر 001 (IT)	طلب الاسقاط	
060102	تصميم المنطق الرقمي	3	3	ح	ث	م	د. رجاء مساحدة	102 (IT)	طلب الاسقاط	
060103	بروتوكولات أمن الشبكات	2	3	ح	ث	م	د. مهدي طحيوش	103 (IT)	طلب الاسقاط	
مجموع الساعات : 13							مجموع ساعات الاسقاط : 3			

اهلا بك : يزن أسامة

رقم المادة	اسم المادة	الشعبة	م.م	الأيام	من	الى	المدرس	القاعة	حصة المادة
060100	أساسيات البرمجة	1	3	ح ت م	12:00	13:00	د. عبدالله الزقية	مختبر (IT) 002	مستفدة
060101	قواعد البيانات	1	4	ح ت م	13:00	14:00	د. مالك المومني	مختبر (IT) 001	
060102	تصميم المنطق الرقمي	3	3	ح ت م	10:00	11:00	د. رجاء مساحدة	(IT) 102	
060103	بروتوكولات أمن الشبكات	2	3	ح ت م	09:00	10:00	د. مهند طحيوش	(IT) 103	
مجموع الساعات : 13									

الخدمات الإلكترونية

المعلومات الشخصية

جدول الطالب

تقديم طلب الاسقاط

الاشعارات (3)

تغيير كلمة المرور

تسجيل الخروج

- Student notification

مرحبًا يزن أسامة

أطلقت	جديد	20:10:20 2024-06-09	قام عميد الكلية بالموافقة على طلب اسقاط مادة أساسيات البرمجة شعبة 1
أطلقت <td>جديد <td>20:10:20 2024-06-09 <td>تم اجراء حركة اسقاط مادة أساسيات البرمجة شعبة 1</td> </td></td>	جديد <td>20:10:20 2024-06-09 <td>تم اجراء حركة اسقاط مادة أساسيات البرمجة شعبة 1</td> </td>	20:10:20 2024-06-09 <td>تم اجراء حركة اسقاط مادة أساسيات البرمجة شعبة 1</td>	تم اجراء حركة اسقاط مادة أساسيات البرمجة شعبة 1
أطلقت <td>جديد <td>20:09:58 2024-06-09 <td>قام رئيس القسم بالموافقة على طلب اسقاط مادة أساسيات البرمجة شعبة 1</td> </td></td>	جديد <td>20:09:58 2024-06-09 <td>قام رئيس القسم بالموافقة على طلب اسقاط مادة أساسيات البرمجة شعبة 1</td> </td>	20:09:58 2024-06-09 <td>قام رئيس القسم بالموافقة على طلب اسقاط مادة أساسيات البرمجة شعبة 1</td>	قام رئيس القسم بالموافقة على طلب اسقاط مادة أساسيات البرمجة شعبة 1
أطلقت <td>جديد <td>20:08:58 2024-06-09 <td>قام المدرس بالموافقة على طلب اسقاط المادة أساسيات البرمجة شعبة 1</td> </td></td>	جديد <td>20:08:58 2024-06-09 <td>قام المدرس بالموافقة على طلب اسقاط المادة أساسيات البرمجة شعبة 1</td> </td>	20:08:58 2024-06-09 <td>قام المدرس بالموافقة على طلب اسقاط المادة أساسيات البرمجة شعبة 1</td>	قام المدرس بالموافقة على طلب اسقاط المادة أساسيات البرمجة شعبة 1
أطلقت <td>جديد <td>03:53:10 2024-06-07 <td>قام رئيس القسم برفض طلب اسقاط مادة أساسيات البرمجة شعبة 1</td> </td></td>	جديد <td>03:53:10 2024-06-07 <td>قام رئيس القسم برفض طلب اسقاط مادة أساسيات البرمجة شعبة 1</td> </td>	03:53:10 2024-06-07 <td>قام رئيس القسم برفض طلب اسقاط مادة أساسيات البرمجة شعبة 1</td>	قام رئيس القسم برفض طلب اسقاط مادة أساسيات البرمجة شعبة 1
أطلقت <td>تم الاطلاع <td>17:02:51 2024-06-08 <td>قام المدرس بالموافقة على طلب اسقاط المادة أساسيات البرمجة شعبة 1</td> </td></td>	تم الاطلاع <td>17:02:51 2024-06-08 <td>قام المدرس بالموافقة على طلب اسقاط المادة أساسيات البرمجة شعبة 1</td> </td>	17:02:51 2024-06-08 <td>قام المدرس بالموافقة على طلب اسقاط المادة أساسيات البرمجة شعبة 1</td>	قام المدرس بالموافقة على طلب اسقاط المادة أساسيات البرمجة شعبة 1
أطلقت <td>تم الاطلاع <td>03:50:16 2024-06-07 <td>قام المدرس بالموافقة على طلب اسقاط المادة أساسيات البرمجة شعبة 1</td> </td></td>	تم الاطلاع <td>03:50:16 2024-06-07 <td>قام المدرس بالموافقة على طلب اسقاط المادة أساسيات البرمجة شعبة 1</td> </td>	03:50:16 2024-06-07 <td>قام المدرس بالموافقة على طلب اسقاط المادة أساسيات البرمجة شعبة 1</td>	قام المدرس بالموافقة على طلب اسقاط المادة أساسيات البرمجة شعبة 1
أطلقت <td>تم الاطلاع <td>16:10:10 2024-06-05 <td>قام المدرس بالموافقة على طلب اسقاط المادة أساسيات البرمجة شعبة 1</td> </td></td>	تم الاطلاع <td>16:10:10 2024-06-05 <td>قام المدرس بالموافقة على طلب اسقاط المادة أساسيات البرمجة شعبة 1</td> </td>	16:10:10 2024-06-05 <td>قام المدرس بالموافقة على طلب اسقاط المادة أساسيات البرمجة شعبة 1</td>	قام المدرس بالموافقة على طلب اسقاط المادة أساسيات البرمجة شعبة 1

الخدمات الإلكترونية

المعلومات الشخصية

جدول الطالب

الاشعارات (5)

تقديم طلب الاسقاط

تغيير كلمة المرور

تسجيل الخروج

Activate Windows
Go to Settings to activate Windows.

- Change password

5.2.3 System Testing Results

The entire system was tested to examine its functionality and quality. The test showed no problems, and the system functions were fully utilized and synchronized as presented in the demo of this project.

5.2.4 Acceptance System Results

After completing our testing phase for this project and implementing the necessary modifications, we have been testing user acceptance through an online survey. The survey is presented in Appendix section (A). It was answered by our system users. Based on the survey results, we determined that the system successfully fulfills all the required functions and possesses user-friendly features, including the ease of use of the main functionality of our system for students, instructors, heads of departments, and deans. These positive findings indicate that the system has met the expectations and preferences of our users, demonstrating its effectiveness and usability.

5.3 Discussion and evaluation

The application works well and there are no loopholes that cause a user to enter an account for another user by mistake and show the expected results in an acceptable manner.

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 Overview

This chapter presents a holistic overview of the project's key aspects, covering its objectives, achievements, main contributions, limitations, and future work.

6.2 Summary of the project

The Course Withdrawal System automates the withdrawal process to streamline and expedite the process for users. This system will enable students to initiate withdrawal requests electronically at any time, track their requests, and receive notifications for any further updates on their request status. Additionally, it allows the student to cancel their pending requests before the instructor's approval.

The instructor receives a notification with every new course withdrawal request to be confirmed or rejected. Both users can also view their schedules, personal information, and provide the ability to change their passwords. This process aims to enhance efficiency, reduce processing times, and provide users with a convenient experience. Through robust technology and secure protocols, our system ensures the safe and reliable transfer of data, empowering users with greater control over their data transactions.

6.3 Achieved objectives

Objective 01: Understand the current project scope and articulate the problem statement comprehensively.

This involves gaining a thorough understanding of the existing situation and challenges faced by students within the manual course withdrawal process, as outlined in Chapter 2. This examination informed decision-making and enabled us to point out the

essential functionalities of our system. Furthermore, we conducted a comparative analysis of our proposed features against those already implemented in existing systems with similar objectives.

Objective 02: Develop and implement the Course Withdrawal System

- **Select A Methodology:**

We chose the scrum methodology due to its flexibility with requirements changing, responsiveness to feedback, and effectiveness with small teams.

- **Determine System Functionalities:**

Defining the system's objectives and features it ought to offer, ensuring alignment with the user needs and expectations.

- **Comparing Functionalities with other Existing Systems:**

To Understand the similar functionalities of other existing systems and determine what needs to be improved in a system with a similar purpose.

- **Doing A Feasibility Study:**

To evaluate Economic, Technical, And Operational aspects, providing essential information to verify the project's sustainability and alignment with goals.

- **Implement Scrum Methodology:**

Breaking the project into multiple sprints to simplify implementation and fit within the time constraints.

- **Designing The System:**

Formulate a comprehensive system design, incorporating various diagrams such as the Context diagram, Data Flow diagram, Use Case diagram, Sequence diagram, the ER (Entity-Relationship) diagram, and Relational diagram.

- **Coding:**

Build the project by using Visual Studio Code and coding in (HTML, CSS, JS, PHP, and Ajax), MySQL Workbench 8.0 CE to build the database, and XAMPP server to execute the code.

Objective 03: System Validation

The validation process involves thorough testing and evaluation of the application to guarantee that it not only meets the specified functional requirements but also effectively addresses the constraints and regulations of the IT college. This phase aims to ensure that the software successfully fulfills its intended purpose.

1. Test Design Specification:

- Login [valid]
- Login using an invalid username or invalid password.
- Click the "Log In" button without providing a username.
- Click the "Log In" button without providing a password.
- Attempting to change the password without providing the correct current password.
- Attempting to change the password without providing 2 identical new passwords.
- View the schedule.
- View the personal information.
- Student attempts to withdraw from an available course in his schedule.
- Student attempts to withdraw from a course without reading/confirming the withdrawal policies.
- Student attempts to withdraw from a course that has been already withdrawn.
- Student attempts to withdraw from a course in which he has been deprived.
- Student attempts to cancel a withdrawal request before the instructor's approval.
- Student attempts to cancel a withdrawal request after the instructor's approval.
- Student attempts to withdraw from a course that has already been rejected.
- Instructor attempts to approve the request or reject it.
- Check notifications.
- Confirm the request or reject it.

2. User Acceptance:

- The system was used by a group of students from the WISE University, mentioned in chapter 5, feedback was systematically gathered through a questionnaire. The outcomes demonstrated that a significant majority of our system's users were impressed by the results.

6.4 Main contributions of the work

The system is designed for WISE university students, with the primary objective of simplifying the course withdrawal procedure. It automates the withdrawal process, enabling students to initiate and complete withdrawals seamlessly. Additionally, it empowers the students to track the progress of withdrawal requests efficiently with an option to cancel the request while it's pending the instructor's approval. This approach aims to save them valuable time and ensure to inform them of all the withdrawal policies.

6.5 Limitation

- The system does not currently support messaging (SMS and Emails) functionality.
- Sometimes the instructor may need to know the reason for a student's withdrawal from a course.
- Potential for Missed Notifications: If users are not logged in they might miss critical notifications.

6.6 Future Work

- **Integration with University System:** The Course withdrawal system has been designed to later be integrated with the existing university system to simplify the course withdrawal process.
- **Implementing messaging functionality:** it would be beneficial to enable this feature. This would allow the instructor, head of department, and dean to easily communicate with students, especially in cases where they need to inquire about the reasons for a student's withdrawal from a course. For withdrawal reasons, the system will be upgraded to enable the student to add the reason for withdrawal in the withdrawal request form.
- **Implement Email Notifications:** Develop and integrate an email notification system that sends updates to student-registered email addresses regarding the status of their withdrawal requests.

REFERENCES

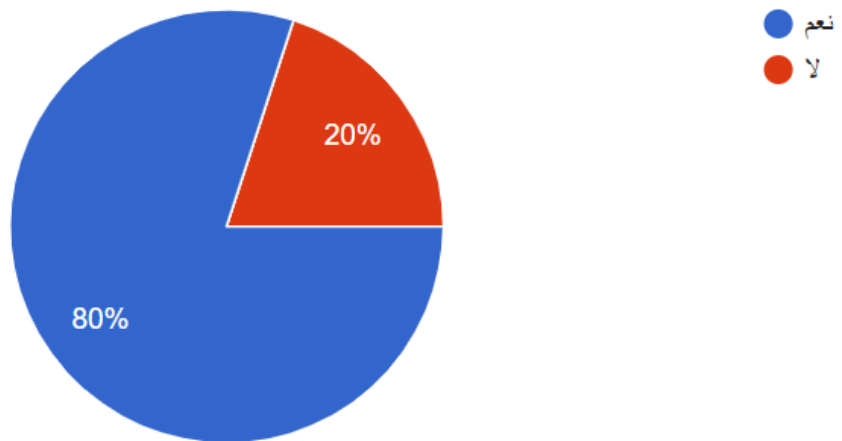
1. Kenneth Kendall & Julie Kendall, Systems Analysis and Design 9e, Prentice Hall.
2. pekka abrahamsson, outi salo, jussi ronkainen & juhani warsta agile software development methods
3. lucidspark ,<https://lucidspark.com/blog/what-is-a-feasibility-study> (accessed April 17, 2024)
4. scrum.org, <https://www.scrum.org/resources/blog/what-sprint-backlog-and-how-it-connects-other-elements-scrum> (accessed May 27, 2024).
5. Ian Sommerville ,SOFTWARE ENGINEERING,9e
6. Miro, ”what’s the context diagram and how do you use it?”,
<https://miro.com/blog/context-diagram/#:~:text=Context%20diagrams%20are%20high%2Dlevel,system%20and%20each%20external%20element.>, (accessed June, 8th 2024)

Appendices

A.

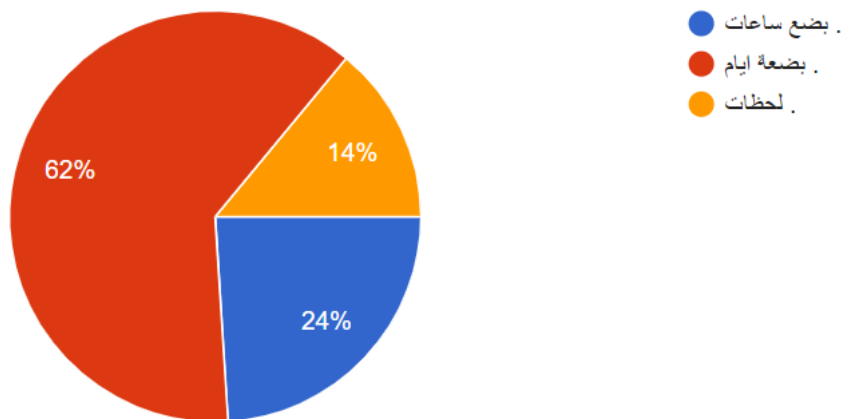
هل قمت باسقاط مادة خلال المرحلة الجامعية؟

50 responses



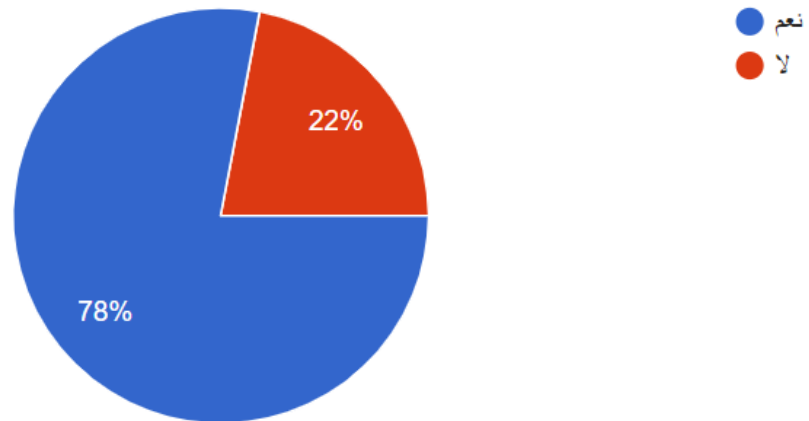
كم استغرقت من الوقت لاتمام طلب الاسقاط ورقيا؟

50 responses



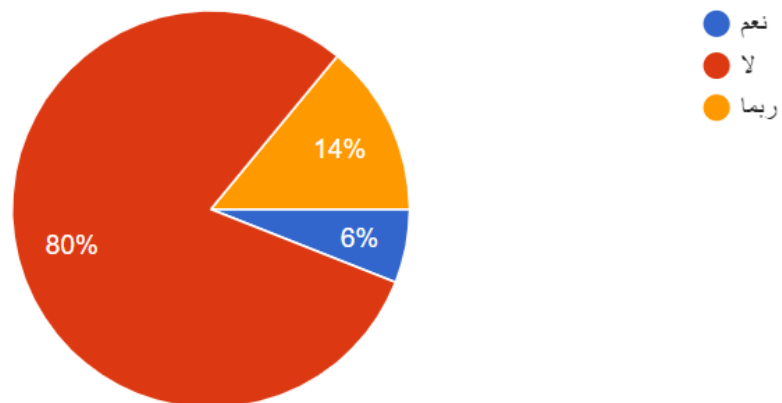
هل كان لديك اي مشكلة في معرفة حالة طلبك ؟

50 responses



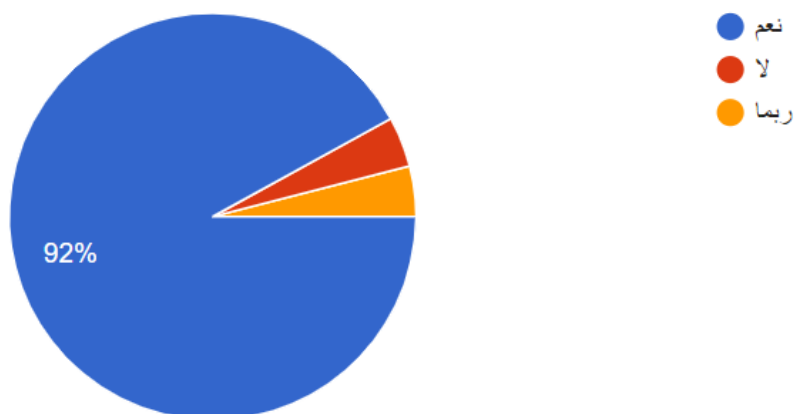
هل كنت تعلم بجميع القوانين المترتبة على اسقاط المواد ؟

50 responses



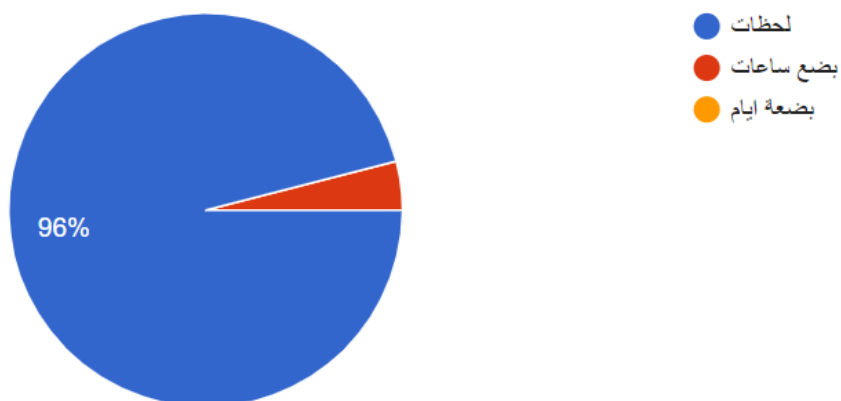
هل عملية الاسقاط الكترونيا وفرت الكثير من الوقت والجهد ؟

50 responses



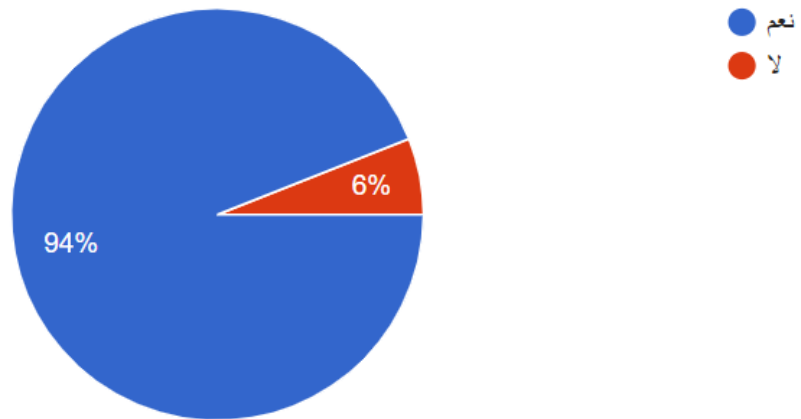
كم استغرق وقت تقديم الطلب الكترونيا؟

50 responses



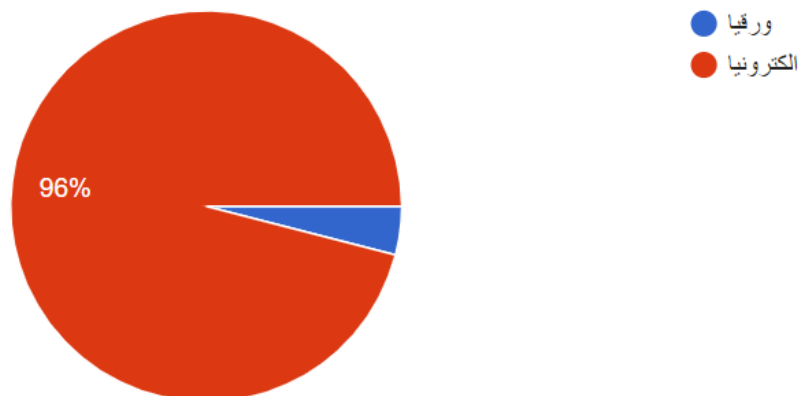
عند تقديم طلب الاسقاط الكترونيا هل تستطيع معرفة حالة طلبك ؟

50 responses



في اي من الطرق كان بإمكانك معرفة حالة طلبك؟

50 responses



عند تقديم طلب الاسقاط الكترونيا هل يتم توضيح جميع التبعيات المترتبة على عملية الاسقاط ؟

50 responses

